Context-Based Aspect-Oriented Requirement Engineering Model

Sonali R. Idate
Department of IT, College of Engineering, Bharati Vidyapeeth (deemed to be University), India
sonaliidate@gmail.com
(corresponding author)

T. Srinivasa Rao
Department of Computer Science, Gitam Institute of Technology, GITAM University, India
sthamada@gitam.edu

Dipak J. Mali
TCS Hinjewadi, India
mali.dipak@tcs.com

Received: 21 January 2023 | Revised: 6 February 2023 | Accepted: 8 February 2023

ABSTRACT

Mobile applications are context-oriented systems that involve the use of context information while operating. Mobile applications demand tackling context information in the early phase of software engineering. A context-aware system demands a different approach to handling the influence of the context on a system's requirements. Aspect-oriented Requirement Engineering separates concerns throughout requirements, called crosscutting concerns, in the early phase of software development to improve the modularity of complex applications. Capturing requirements embedded within context is a challenging procedure. This study aimed to identify such contextual characteristics of requirements in the early phase of software engineering, using natural language processing techniques, by proposing Context-Based Aspect-Oriented Requirement Engineering (CB-AORE) to visualize the existence of crosscutting concerns. CB-AORE performs context modeling to analyze the context dependency with base requirements and helps the analyst to visualize the correlation of functional and non-functional requirements with context. A case study analyzed the identification of context and its use to identify crosscutting concerns.

Keywords-context; context-aware system; crosscutting concern; aspect; aspect-oriented requirement engineering

I. INTRODUCTION

Requirement engineering is one of the important phases in software engineering, which includes a set of functional requirements specifying what functions a system should perform and a set of non-functional requirements specifying the performance characteristics of the functional features. Functional requirements present the operational view to both the client and the developer and must be satisfied in terms of ease of use and less cost. Due to their context-sensitive nature, mobile applications have some unique requirements that are less common in traditional software applications [1]. These requirements, including sensor handling, potential interactions with other applications from different sources, and embedded hardware devices [2], increase the complexity of mobile application development. Software engineering analysis specifies what the system is currently doing, and design specifies what a new or modified system will do [3]. System design includes a top-down approach, coupling, cohesion, span of control, size, and sharing of modules. The top-down approach means understanding the problem at the top level and trying to exploit it at lower levels, which is also known as modularization. Object-Oriented Software Development (OOSD) handles the complexity of an application using modularization and abstraction [4]. In [5], a formal aspect-oriented requirement specification language was proposed, traced with the structural model, for complex system analysis. In [6], a role-based requirement modeling technique was proposed to facilitate analysis in the development of large and complex systems involving a large number of stakeholders.

OOSD aims to provide independence between modules, improve reusability, faster development, and reduced cost. However, Aspect-Oriented Software Development (AOSD) does these in a much better way [7] and is possible by separating concerns. Aspect-Oriented Programming (AOP) introduces a new term called aspect, which represents crosscutting concerns as a module [8]. Usually, concerns refer to the functionality offered by the system. AOP can be used for complex cryptographic modeling and security simulation without changing legacy code [9]. Aspect-Oriented Requirements Engineering (AORE) is an early phase in AOSD that identifies crosscutting concerns at the level of requirements [10]. The process of separating concerns is called crosscutting concerns [11, 12], which is required in applying AOSD methods.

This paper proposes a requirement modeling process using additional context-specific requirements from a software requirement specification document and identifies crosscutting functional and non-functional requirements. Engineering context-aware software systems, such as mobile applications, face many challenges [13]. Context is any information that can be used to characterize the situation of an entity, such as a person, place, or object, necessary for the interaction between the user and the application [14]. This model was inspired by the multidimensional approach for separating concerns and identifying homogeneous crosscutting concerns [15-16]. Homogeneous cross-cutting concerns are concerns that add the same behavioral variation at multiple points in the program [11]. Moreover, the multi-dimensional approach treats all concerns at the same level and visualizes them from meta-concern and system spaces. A contribution matrix, developed from the compositional intersection of concerns, provides trade-off points that help to make architectural choices for the system [15]. However, the multidimensional approach lacks explicit relationship interaction [17].

The proposed Context-Based Aspect-Oriented Requirement Engineering (CB-AORE) method provides immediate traceability at a high level of design. The future scope of CBAORE specifies the articulation of the dependency relationship structure. This model contends that context-relevant data must be gathered from the standard set of requirements. There is a need for decomposing requirements based on context to identify context-based functional and non-functional requirements. The aim of the paper is twofold: classify typical requirements into functional and non-functional and then apply the proposed CB-AORE technique to identify crosscutting concerns in the early phase of software development.

II. METHODOLOGY

The proposed model used context-related requirement elements by studying requirements less commonly found in traditional software [1]. This was a two-fold activity. At first, a survey was conducted with a sample questionnaire to identify the need for a context-based model for requirement engineering. The second activity primarily focused on the extraction of mobile applications' specific requirements exhibiting special characteristics related to context. These requirements were extracted from 20 requirements documents for various mobile applications. The various contexts identified in the literature and SRS documents of earlier projects were Sensors Context, Device Context, Transactional Context, Network Context, Mobility Context, Social Context, and Third Party Software Integration [18]. The special requirements for mobile applications were the use of sensors for human interactions, the use of embedded devices, and various transactions using a network. Based on special requirements, the sample questions to investigate the need for context categories, relevant context elements, and activity of the mobile application are:

- Q1: Do you find that requirement analysis based on context consideration in mobile applications will help programmers?
- Q.2: Do you find the context-based requirement model understandable?
- Q3: Were you able to identify individual functional requirements and their relevant context?
- Q4: Do you agree that the implementation of mobile context-sensitive applications can be facilitated by the context-based requirement model?
- Q5: Do you think that the development of mobile context-aware applications can be facilitated by a model that enables the automatic production of classes and methods?
- Q6: Do you agree that exposing the context dependencies with the functional requirements makes it easier to explain to a business user of the app?
- Q7: How does the application adjust to the context?
- Q8: Do you find interdependency in the context?
- Q9: Were you able to identify non-functional requirements from context?

As there is not a specific set definition for context, everyone can provide his perspective on it, which can be categorized into different classes. The proposed model was inspired by the multidimensional approach to the separation of contexts in requirement analysis, which applies uniform treatment to different types of requirements [17]. The proposed CB-AORE model overcame the drawbacks of the multidimensional approach, such as exhibiting a lack of explicit relationship interaction among different requirements. Seven context dimensions were identified and were embedded in the CB-AORE model, as shown in Table I.

TABLE I. CONTEXT CATEGORIES

Context	Context elements	Activity description
Device context	Camera, microphone, speaker, battery	Accessing camera, microphone, speaker, battery, device infrastucture/facilities such as time,date etc.
Sensor context	Accelerometer, proximity	Accessing sensors functionality/services
Transactional context	Internal storage, External storage (cloud)	Accessing data/information on/off mobile device, modify, add, delete data with I/O
Network context	Internet, wifi, Bluetooth	Accessing web server through Internet connection or Bluetooth
Mobility context	GPRS, GPS	Accessing map and location and provide navigation facility
Social context	Sharing, notification	Sharing information among users through social networks
Third party Software Integration	API	Integrating app with third party services such a newsfeed service, payment gateways etc.

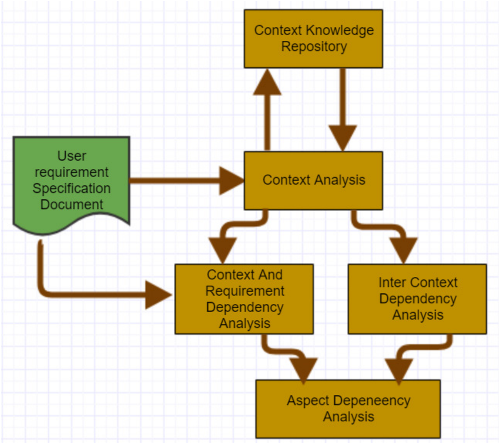


Fig. 1. Engineering model of CB-AORE.

These dimensions are:

- Device context: Contains mobile device features that allow users to access information about any web domain area. The features of a mobile device can be broken down into its technological, functional, and physical attributes [19-20]. Physical attributes include weight, overall size, screen size, and resolution [21], functional attributes include input processes, gestures, and output mechanisms, technological attributes include CPU speed and storage capacity, and functional attributes include elements such as camera, microphone, etc. [22].
- Sensor context: Mobile e-health apps use a variety of sensors, including an accelerometer, a low-pass filter, and a magnitude filter, to track the body's diabetes levels while the user is moving, walking, or running [23-24]. The motion, posture, and location of a mobile device can be identified by the 3D accelerometer, digital compass, etc. Dedicated context-aware frameworks were developed to integrate sensor data, complex events, and data storage [25].
- Transactional context: It is important for taking any kind of input data and analyzing them [26]. Such mechanisms transfer data to and from internal or external storage.
- Network context: Characterizes the ability of a wireless mobile device to move from one area to another while still being able to access the data connection network facility from anywhere inside the network zone [24]. The present, growing wireless network technologies and mobile applications depend on it in fundamental ways.
- Mobility context: Applies to the attributes and functionalities of mobile applications that use network and location elements [27].
- Social context: Characterizes the activity that involves the exchange of information from one user to another through social media [19]. This context determines social relations through social media tools such as wikis, content hosting, social networking platforms, blogs, and podcasting.
- Third-Party Software Integration (TPSI) context: Captures elements of mobile apps and devices on the integration of extended functionality deployed on multiple platforms [21].

III. THE CB-AORE MODEL

In software engineering, a software requirement specification document represents requirements at a higher level of abstraction [3]. Context analysis is performed on user requirements with the help of a context repository. Then, context interdependency analysis activities are mainly carried out by using syntactic analysis with natural language processing techniques. Further dependency analysis of context with requirements is used to identify dependencies among aspects. Dependency analysis is carried out by establishing dependency links among contexts and between contexts and requirements. Contexts are extracted and analyzed from textual requirements. However, this requires understanding the effects of context on different requirements with reasoning [28]. Therefore, the context model can be expressed as follows:

- Definition 1: A Context Model is 2-tuple CM = (C, DL), where:

  C is a finite set of contexts, $C \neq \Phi$;

  $DL \subset (C \times C)$ is a set of dependency links.

- Definition 2: c is one context of the set C with the following properties:

1. *c.dec* represents the decomposition of context based on functional and non-functional requirements.
2. *c.level* represents the level of context. The value set of *c.level* is {*user, system*}, representing that c is *user* or *system* context, respectively.
3. *c.link* represents whether *context* c has links to FR, NFR, and other contexts. The value set of *c.link* is {1,0}.
4. *c.type* represents the type of context. The value set of *c.type* is {Sensor context, System context, Transactional context, Network context, Mobility context, Social context, Third party Software Integration, GUI, Search, Misc}

The potential relationship and dependency links are conceptualized using the CB-AORE model. The model is represented with the help of ROADMAP and I* notations, as shown in Tables II and III. Table IV shows sample sets of text requirements from the inSEPtion mobile application. This mobile application was developed to guide students who have just arrived on campus.

TABLE II. ROADMAP NOTATIONS

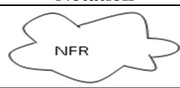
Notation	Role description
	Nonfunctional requirement
	Context
	Role

TABLE III. I* NOTATIONS

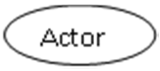
Notation	Role description
	Context
	Nonfunctional requirement
	Actor
	Context dependency

TABLE IV. REQUIREMENT TEXT

R1	The user should be able to search for a certain point of interest and navigate from a specific location to another.
R2	The application should provide a map view of the campus.
R3	The application shows all upcoming events regarding the university as provided in an RSS feed
R4	Any event should be presented in a calendar form and should be searchable.
R5	The application shows news items regarding the university as provided in an RSS feed.
R6	The application enables the student to read the student handbook
R7	Facebook link directs the students to the facebook page of the university/ability to view other facebook pages related to university/ furthermore uses facebook functionalities such as 'Like', 'Share'.
R8	Twitter application directs student to the twitter page of the university.
R9	The user can search for personnel information available on the university server.

NLP techniques were applied in stages for context realization. Context repository extracted the context categories from the requirements text. In the second step, context initiation either by the system or user was identified. Context dependencies with non-functional requirements are established through text processing and mapping. The mapping function of any subject clause x in non-functional requirements that matches the vocabulary of a context category is:

$$f_{map(NF_m, C_n)}(x) = 1 \quad (1)$$

where m is the number of the non-functional requirement and n is the number of the context category. Similarly, for functional requirements:

$$f_{map(F_m, C_n)}(x) = 1 \quad (2)$$

Requirements R1 to R9 are either initiated by the user or the system. The "view map" functionality in R1 requires the network context, so the mobility context depends on it. In R9, when accessing personnel information from a server, the transactional context depends on the networking context. Functional requirements are passed through NLP processes to identify contexts initiated by actors. As shown in the diagram, mobility context is captured from activity descriptions such as 'search location' or 'navigate' in R1. Similarly, R2 realizes mobility context. However, context is initiated by the user in the first case and by the system in the latter.

Figure 2 shows context realization through context elements for the actor. R1 and R2 depend on mobility context. Mobility context depends on network context, as it requires a data connection context element for map view. To provide updated news, the system depends on TPSI which is possible using an API context element and the network context's Internet context element. Every context element specifies a nonfunctional requirement so that realization can successfully satisfy the requirement. In this scenario, to provide a map view, network context must provide a network context element with the required speed and bandwidth, i.e. reliability and availability. Likewise, all requirements are mapped with their related context element.

Figure 3 shows the dependency between a context element and a functional requirement. A context element related to a functional requirement exhibits the latter's dependency on the context. For example, 'R8: redirect to Facebook' and 'R7: redirect to Twitter' are functional requirements related to social media activity and depend on Wi-Fi or the internet connection context element. Figure 3 visualizes the dependencies of the functional requirements R7 and R8 with the social context and the network context. Thus, context elements through specific services fulfill functional requirements and also specify the needed non-functional requirements to successfully execute it.

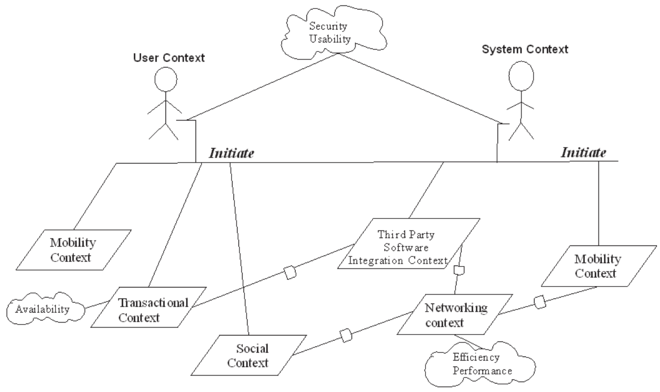


Fig. 2. Context and nonfunctional requirement dependencies.

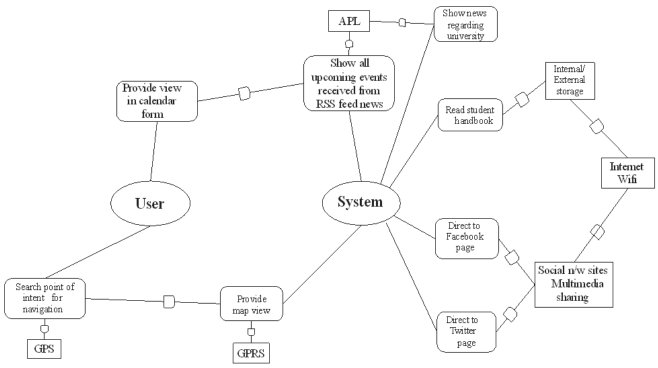


Fig. 3. Context and functional requirement dependencies.

IV. COMPARATIVE ANALYSIS OF CB-AORE WITH OTHER AORE MODELS

Numerous methods have been proposed to identify early aspects using different techniques. In [29], an extensive comparative analysis on the Theme/Doc approach and Multidimensional Separation of Concerns for Requirements Engineering (MDSOCRE) was performed, concluding that MDSOCRE focuses on the analyst's domain knowledge, and Theme/Doc analyses the requirements language while looking for the concern name [29]. The performance of the proposed method was verified by comparing it with goal-based AORE [16], viewpoint-based AORE [1], and multidimensional separation of concerns in requirements engineering [15] methods. Table V shows a brief comparison.

TABLE V. CB-AORE COMPARISON WITH OTHER METHODS

Parameters	**CB-AORE**	**GEA Goal based [16]**	**PREView View point based [1]**	**Multi-Dimensional Separation of Concerns in Requirements Engineering [15]**
Applicability for mobile application	Yes	No	No	Yes
Requirement modeling	Mobile context-based	Goal and soft goal based	Viewpoint based	Multi-dimensional
Composability	Per context element and requirement	Per goal and softgoal	Per view	Per meta concern space and system space
Handling of functional and non-functional concern	Both	Functional	Non-functional	Both
Stakeholder scope	Mobile device, developer, system, and user	Analyst, developer		Not applicable
Methodology	Natural language analysis	Fuzzy logic, clustering technique	Fuzzy logic	XML-based composition specification language to specify the composition rules

A detailed comparison highlights the following:

- Use of requirement elicitation artifact: Goal-based AORE uses a case model by analyzing goal interaction, where the structure of goals considers various facets of requirements. The viewpoint-based approach divides requirements using a viewpoint to derive candidate aspects in the analysis phase. The proposed CB-AORE model relies on functional and non-functional requirements from any requirement document and realizes context by mapping the requirements with the context repository.
- Candidate early aspect realization technique: In the Goal-based approach, early aspects are identified by grouping goal interactions using a clustering algorithm. In the viewpoint approach, each candidate's early aspect is qualified with the help of concerns that are non-functional requirements. A candidate's early aspect is the concern that cuts across multiple viewpoints. In CB-AORE, context is the candidate aspect that cuts across functional and non-functional requirements.
- Software application domain: The Goal-based approach targets software system applications that have large concentration of only functional and nonfunctional characteristics to describe the requirements of the application. The Viewpoint-based approach also relies on the functional and non-functional characteristics of any software system. So both approaches fail to accommodate context characteristics exhibited by mobile software applications. The CB-AORE approach helps to envision the interactions not only between functional and non-functional requirements but also context.

V. DISCUSSION

Many initiatives have been proposed to collect and relate aspect-oriented requirements in the early stages of software development [4, 11-12, 16, 29-30]. However, the proposed method decomposes the requirements based on contexts, which are special characteristics of a mobile application's functional and non-functional requirements. The further decomposition of contextual requirements into heterogeneous and homogeneous was not considered. Furthermore, it would be interesting to focus on tracking contextual aspects in architectural decisions.

VI. CONCLUSION

This paper proposed the possibility of using context in mobile applications to derive dependencies among functional and non-functional requirements in the early phase of requirement engineering. Requirement engineering of mobile applications can utilize context characteristics to achieve better modularity for developing mobile software applications using an emerging AOSD paradigm. A case study of the presented CB-AORE gave an insight into how context dependencies of functional and non-functional requirements can identify crosscutting concerns and early aspects. Future work should articulate dependency structure for describing relationships among context, functional, and non-functional requirements and also aim to strengthen the composition ability of aspects.

REFERENCES

[1] A. I. Wasserman, "Software engineering issues for mobile application development," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, Santa Fe, NM, USA, Aug. 2010, pp. 397–400, https://doi.org/10.1145/1882362.1882443.

[2] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Technical Report TR2000-381, Nov. 2000. [Online]. Available: https://digitalcommons.dartmouth.edu/cs_tr/177.

[3] P. Sawyer, I. Sommerville, and S. Viller, "PREview: Tackling the Real Concerns of Requirements Engineering," Cooperative Systems Engineering Group, Technical Report CSEG/5/1996, 1996.

[4] A. Rashid, P. Sawyer, A. Moreira, and J. Araujo, "Early aspects: a model for aspect-oriented requirements engineering," in *Proceedings IEEE Joint International Conference on Requirements Engineering*, Essen, Germany, Sep. 2002, pp. 199–202, https://doi.org/10.1109/ICRE.2002.1048526.

[5] C. L. Vidal-Silva, E. Madariaga, T. Pham, F. Johnson, L. A. Urzua, and L. Carter, "JPIAspectZ: A Formal Requirement Specification Language for Joint Point Interface AOP Applications," *Engineering, Technology &*

Applied Science Research, vol. 9, no. 4, pp. 4338–4341, Aug. 2019, https://doi.org/10.48084/etasr.2774.

[6] C. W. Shiang, A. A. Halin, M. Lu, and G. CheeWhye, "Long Lamai Community ICT4D E-Commerce System Modelling: An Agent Oriented Role-Based Approach," *The Electronic Journal of Information Systems in Developing Countries*, vol. 75, no. 1, pp. 1–22, 2016, https://doi.org/10.1002/j.1681-4835.2016.tb00547.x.

[7] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. New York, NY, USA: McGraw-Hill, 2005.

[8] O. A. Abdulhameed, A. Y. Yousuf, and R. H. Abbas, "Aspect oriented programming: Concepts, characteristics and implementation," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 7, no. 4, pp. 2022–2033, Jan. 2020, https://doi.org/10.21533/pen.v7i4.975.

[9] H. Mestiri, I. Barraj, and M. Machhout, "AES High-Level SystemC Modeling using Aspect Oriented Programming Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 1, pp. 6719–6723, Feb. 2021, https://doi.org/10.48084/etasr.3971.

[10] M. Springer, "A comparison of Context-Oriented and Aspect-Oriented Programming," *Seminar Paper*, 2014.

[11] A. Razzaq and R. Abbasi, "Automated separation of crosscutting concerns: Earlier Automated identification and modularization of cross-cutting features at analysis phase," in *2012 15th International Multitopic Conference (INMIC)*, Islamabad, Pakistan, Sep. 2012, pp. 471–478, https://doi.org/10.1109/INMIC.2012.6511500.

[12] S. Apel, D. Batory, and M. Rosenmuller, "On the Structure of Crosscutting Concerns," presented at the GPCE Workshop on Aspect-Oriented Product Line Engineering, Portland, OR, USA, Oct. 2006.

[13] S. Rubab, B. Dhupia, B. Jaafar, and N. Litayem, "Investigating user Requirements for Mobile Educational App Impact of Requirements Gathering on Software Development," *International Journal of Engineering Research & Technology*, vol. 4, no. 3, Mar. 2015, https://doi.org/10.17577/IJERTV4IS030604.

[14] L. Han, S. Jyri, J. Ma, and K. Yu, "Research on Context-Aware Mobile Computing," in *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, Gino-wan, Japan, Mar. 2008, pp. 24–30, https://doi.org/10.1109/WAINA.2008.115.

[15] A. Moreira, A. Rashid, and J. Araujo, "Multi-dimensional separation of concerns in requirements engineering," in *13th IEEE International Conference on Requirements Engineering (RE'05)*, Paris, France, Dec. 2005, pp. 285–296, https://doi.org/10.1109/RE.2005.46.

[16] J. Lee and K.-H. Hsu, "GEA: A Goal-Driven Approach toDiscovering Early Aspects," *IEEE Transactions on Software Engineering*, vol. 40, no. 6, pp. 584–602, Jun. 2014, https://doi.org/10.1109/TSE.2014.2322368.

[17] M. L. Koole, "A Model for Framing Mobile Learning," in *Mobile Learning: Transforming the Delivery of Education and Training*, Edmonton, AB, Canada: Athabasca University Press, 2009, pp. 25–49.

[18] G. Kiczales *et al.*, "Aspect-oriented programming," in *ECOOP'97 — Object-Oriented Programming*, Jyväskylä, Finland, 1997, pp. 220–242, https://doi.org/10.1007/BFb0053381.

[19] K. C. Kamani and D. R. Kathiriya, "Cultivate ICT & Networking: the Role of social media in agriculture," *CSI Communications*, vol. 37, no. 7, pp. 15–17, 2013.

[20] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Big Island, HI, USA, Jan. 2003, https://doi.org/10.1109/HICSS.2003.1174831.

[21] B. Bender, "The Impact of Integration on Application Success and Customer Satisfaction in Mobile Device Platforms," in *Platform Coring on Digital Software Platforms*, B. Bender, Ed. Wiesbaden: Springer Fachmedien, 2021, pp. 79–118.

[22] J. Kolari *et al.*, "Context-Aware Services for Mobile Users: Technology and User Experiences," VTT Technical Research Centre of Finland, Espoo, 2004.

[23] D. Preuveneers, Y. Berbers, and W. Joosen, "The Future of Mobile E-health Application Development: Exploring HTML5 for Context-aware Diabetes Monitoring," *Procedia Computer Science*, vol. 21, pp. 351–359, Jan. 2013, https://doi.org/10.1016/j.procs.2013.09.046.

[24] A. R. Lamas, J. L. Filho, A. de Paiva Oliveira, and R. M. de Almeida Botelho Junior, "A Mobile Geographic Information System Managing Context-Aware Information Based on Ontologies," *UbiCC Journal*, vol. 4, pp. 718–727, 2009.

[25] E. M. Milic and D. Stojanovic, "EgoSENSE: A Framework for Context-Aware Mobile Applications Development," *Engineering, Technology & Applied Science Research*, vol. 7, no. 4, pp. 1791–1796, Aug. 2017, https://doi.org/10.48084/etasr.1203.

[26] J. Floch, S. Hallsteinsen, A. Lie, and H. I. Myrhaug, "A Reference Model for Context-Aware Mobile Services," SINTEF Telecom and Informatics, Trondheim, Norway, 2001.

[27] Q. H. Mahmoud, "Provisioning Context-Aware Advertisements to Wireless Mobile Users," in *2006 IEEE International Conference on Multimedia and Expo*, Toronto, ON, Canada, Jul. 2006, pp. 669–672, https://doi.org/10.1109/ICME.2006.262534.

[28] T. Yamabe and T. Nakajima, "Possibilities and Limitations of Context Extraction in Mobile Devices: Experiments with a Multi-sensory Personal Device," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 4, no. 4, pp. 37–52, Oct. 2009.

[29] A. Zambrano, J. Fabry, and S. Gordillo, "Expressing aspectual interactions in requirements engineering: Experiences, problems and solutions," *Science of Computer Programming*, vol. 78, no. 1, pp. 65–92, Nov. 2012, https://doi.org/10.1016/j.scico.2011.12.004.

[30] T. Elrad, O. Aldawud, and A. A. Bader, "Expressing Aspects Using UML Behavioral and Structural Diagrams," in *Aspect-Oriented Software Development with Use Cases (Addison-Wesley Object Technology Series)*, Boston, MA, USA: Addison-Wesley Professional, 2004, pp. 459–478.